



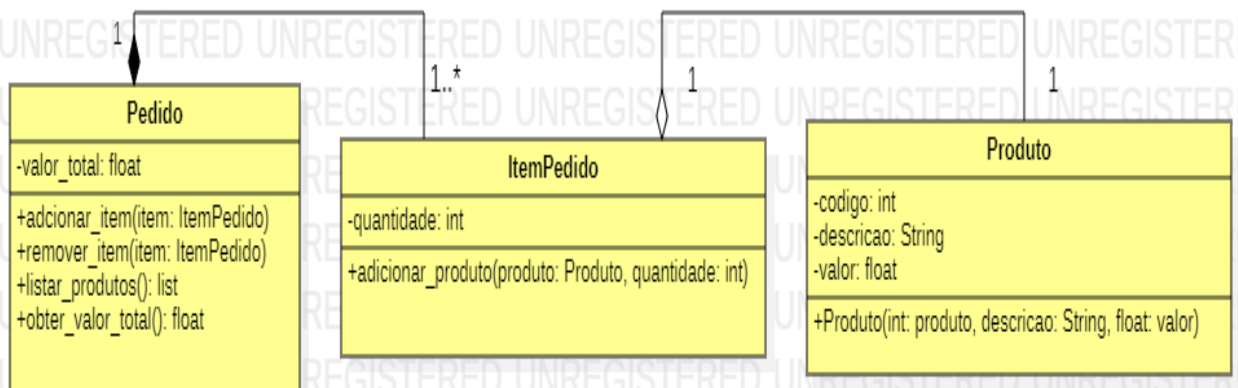
INSTRUÇÕES

- Responder às questões abaixo utilizando os conceitos aprendidos sobre a linguagem de programação Python;
- Criar e colocar no GitHub o diagrama de classes na ferramenta StarUML ou similar;
- Enviar o código fonte no repositório da turma no GitHub criando uma pasta com o nome da dupla que respondeu as atividades;
- Separe cada exercício em uma pasta separada;
- Cada classe deverá estar num arquivo com o mesmo nome (exemplo: classe Pessoa no arquivo pessoa.py);
- Para cada exercício crie um arquivo principal (main.py) que irá importar as classes que são utilizadas;
- O arquivo main.py será o único que irá receber dados de entrada do usuário (input) e imprimir dados de saída (print).

EXERCÍCIO DE FIXAÇÃO PYTHON

1. Implemente as classes descritas no diagrama abaixo, onde:

- Um pedido é composto por um conjunto de itens pedidos.
- Um item pedido associa-se com um ou vários produtos.
- O cálculo do valor total do pedido deverá ser feito mediante a soma do preço de cada produto incluído nos itens pedidos.
- A classe Produto não precisa dos métodos Setter e Getter, basta implementar um construtor com parâmetros nomeados. Os produtos serão todos criados na classe main.py usando esse construtor.
- A classe ItemPedido também não precisará de métodos Setter e Getter, ao invés disso, no método solicitar_produto ele deverá ser preenchido com a quantidade desejada no pedido do cliente (sendo esse o único caminho para alterar o valor de quantidade).
- A classe Pedido deverá adicionar, remover, listar e mostrar o valor total do pedido.
- Sobre a multiplicidade:
 - 1 pedido deve conter nenhum ou vários itens pedidos;
 - 1 item pedido está associado a 1 produto.



2. Crie um diagrama de classes que represente uma classe Revista com os atributos privados código, título, tipo, edição, e uma classe Edição contendo os atributos privados número da edição, data da edição e número de artigos. Lembre-se que uma revista “contém” várias edições e implemente métodos para representar esse relacionamento.
 - Cardinalidades:
 - Uma Revista contém uma ou várias edições (1 para muitos).
3. Crie um diagrama de classes que represente uma classe Pessoa com os atributos privados identificador, nome e CPF, e uma classe Endereço com os atributos número da casa, rua, cidade, estado e país. Nesse caso uma pessoa deve “agregar” um ou vários endereços. Implemente métodos para representar esse relacionamento.
 - Cardinalidades:
 - Uma Pessoa agrega um ou vários endereços (1 para muitos).